State transformation in event driven explicit simula

Nguyen, T.V. Devgan, A.
Res. Lab., IBM Corp., Austin, TX, USA ;

This paper appears in: **Computer-Aided Design, 1997. Digest of Technic 1997 IEEE/ACM International Conference on**

Meeting Date: 11/09/1997 - 11/13/1997
Publication Date: 9-13 Nov. 1997
Location: San Jose, CA USA
On page(s): 289 - 294
Reference Cited: 7
Number of Pages: xxvi+767
Inspec Accession Number: 5781434

Abstract:
This paper presents a general method for incorporating **state** transformation **driven** explicit **simulation**. One inherent assumption in this type of **simulati** algorithm is the state independence, which allows the algorithm to process th independently in an **event driven** manner at the transistor level. Numerical p arise when an inappropriate state representation of the **circuit**, in which the s not truly independent, is chosen. In principle, any similarity transformation of equation can be employed to transform the **circuit** into a more convenient st for numerical solution. This paper develops a systematic scheme to derive an state transformation, and to incorporate the state transformulation in such a maintain the efficiency of **event driven** explicit **simulation** algorithms

Index Terms:
circuit analysis computing discrete event simulation event driven explicit simulatio transformation simulation algorithm state equation state representation state trans

Documents that cite this document
There are no citing documents available in IEEE Xplore at this time.

Publications | Help | FAQ| Terms | Back to Top

State Transformation in Event Driven Explicit Simulation

Tuyen V. Nguyen and Anirudh Devgan
IBM Austin Research Laboratory
Austin, TX 78758
tuyenn;devgan@austin.ibm.com

Abstract

This paper presents a general method for incorporating state transformation in event driven explicit simulation. One inherent assumption in this type of simulation algorithm is the state independence which allows the algorithm to process the states independently in an event driven manner at the transistor level. Numerical problems arise when an inappropriate state representation of the circuit, in which the states are not truly independent, is chosen. In principle, any similarity transformation of the state equation can be employed to transform the circuit into a more convenient state space for numerical solution. This paper develops a systematic scheme to derive an appropriate state transformation, and to incorporate the state transformation in such a way to maintain the efficiency of event driven explicit simulation algorithms.

1. Introduction

With design trends continuously moving towards higher clock speed, higher density, and higher levels of integration on chip, the need for fast circuit simulation of large and complex integrated circuits is well known. There have been two major approaches in improving the efficiency of (circuit) transient simulation. Common to all approaches is the use of circuit partitioning and event driven simulation to exploit the spatial sparsity and the temporal latency of large circuits. One approach is to simplify circuit topology to a set of primitives, especially for MOS circuits, in addition to employing simplified device models, such as piecewise quadratic models, to derive the analytical solutions of the node voltages. This approach is embodied in the simulator ILLIADS [3]. The other major approach employs simplified device models to speed up the computation of the numerical solutions during a transient analysis. Piecewise constant (PWC) device models are employed in SPECS [4] and ADAPTS [5]. More general PWL models have been used in various timing analyzers and simulators [1][6]. The Adaptively Controlled Explicit Simulation (ACES) algorithm [2], has been developed to exploit the efficiency of explicit integration algorithm while ensuring the numerical stability of the integration at an incremental cost instead of the full cost of an implicit algorithm. The basic idea is to incrementally approach the steady state while ensuring the stability and efficiency of the integration algorithm.

In event driven simulation, an implicit assumption is that the states are independent so that they can be processed separately. If some of the states are strongly coupled or dependent, these states have to be processed as a group. In other words, the choice of the state variables for a state representation of a given circuit is not appropriate in the sense that the chosen states are not truly independent. This situation arises quite often in extracted circuits, which contain numerous parasitic resistors of small values. For example, given a resistor as shown in Fig. 1, the two capacitors at the two

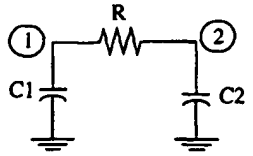

Fig. 1: *A simple illustration of the state dependency problem associated with a small value resistor*

nodes of the resistor are usually chosen to be the state variables in a typical circuit formulation such as Modified Nodal Analysis (MNA). In the degenerate case of a zero valued resistor or voltage source, the two states are actually the same. Even if the value of the resistance is very small, i.e. the two nodes close to being shorted, then the two chosen state variables are not truly independent. In the following, a number of numerical problems associated with the case of small resistors mentioned above are discussed.

Matrix ill-conditioning

The first one is the numerical ill conditioning problem due to the resistor being stamped into the MNA circuit matrix equation as a conductance term with $G = 1/R$ in the admittance form

$$i_R = Gv_R = G(v_1 - v_2).$$

The reason that resistors are stamped as conductors is to avoid introducing extra unknowns into the MNA matrix equation, in which the capacitor voltages and inductor currents are chosen as state variables. However, if R is small, then G is large and it may degrade the numerical conditioning of the matrix equation. One way to avoid this problem is to stamp the resistor in its impedance form as

$$v_R - Ri_R = v_1 - v_2 - Ri_R = 0$$

Note that this impedance form introduces the extra unknown current variable i_R into the matrix equation. Note further that this stamping does not address the state dependency problem though it may alleviate the numerical conditioning problem.

State Dependency in Explicit Simulation

The state dependency problem is evident in explicit integration algorithm such as Forward Euler (FE). Given the Branch Constitutive Relation (BCR) of a capacitor as $i_C = C\dot{v}_C$, the FE stamp of the capacitor at each time t is given by

$$v_C(t) = v_C(t - \Delta t) + \frac{\Delta t}{C} i_C(t - \Delta t)$$

where $t - \Delta t$ is the previous time point and Δt is the time step. The above equation represents the BCR of a voltage source, and the

equivalent circuit of the simple structure shown in Fig. 1 using FE integration is shown in Fig. 2 below. As illustrated by Fig. 2, if R is

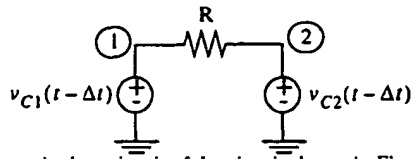

Fig. 2: *The equivalent circuit of the circuit shown in Fig. 1 at time t using an explicit (Forward Euler) integration for the capacitors.*

small or zero in the degenerate case, the equivalent circuit includes an illegal loop of voltage sources. This state dependency problem will affect any explicit integration algorithm that simply update the capacitor voltages at a given time using only values at previous time points.

State Dependency in Implicit Simulation

The problem of state dependency is handled automatically in an implicit integration scheme because all the states are "implicitly" processed together at the same time. For example, using Backward Euler (BE) as the implicit integration scheme, the stencil of a capacitor in the MNA matrix equation at each time t can be written as

$$v_C(t) = \frac{\Delta t}{C} i_C(t) + v_C(t-\Delta t) = G_{eq} i_C(t) + V_{eq}$$

In circuit theoretic terms, this equation represents an equivalent circuit consisting of a resistor with conductance $G_{eq} = (\Delta t)/C$ in series with a voltage source $V_{eq} = v_C(t-\Delta t)$. Using these equivalent circuits for the two capacitors, the circuit shown in Fig. 1 can be redrawn as shown in Fig. 3 below. In this case, even if $R = 0$, the

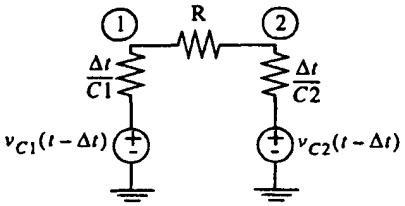

Fig. 3: *The equivalent circuit of the circuit shown in Fig. 1 at time t using an implicit (Backward Euler) integration for the capacitors*

two equivalent voltage sources are still decoupled by the two equivalent conductors, and these voltage sources do not form an illegal loop.

The fundamental problem is then how to handle this state dependency problem without resorting to implicit integration algorithms for efficiency reasons. One simple solution is to preprocess a given circuit and delete from the circuit description all the resistors with small values below a specified tolerance. However, this approach may require setting different tolerances for different circuit configurations. Moreover, in some cases, it may be necessary to retain these small resistors in the circuit description. For example, given a long RC interconnect with many taps along the interconnect, it is necessary to retain the small resistors in each RC section of the interconnect to get the total resistance loading of the line. This problem obviates the need to develop an algorithmic approach, other than circuit pruning, to handle this problem. It is the goal of this paper to present such an approach using the event driven Adaptively Controlled Explicit Simulation (ACES) algorithm as an implementation vehicle. This paper is organized as follows. In section 2 of the paper, a simple example is presented to review the basic ACES algorithm as an implementation vehicle and illustrate the numerical problems due to state dependency. The main results of the paper are presented in section 3. This section discusses a general and systematic method to incorporate state transformation into event driven explicit simulation, especially its implementation in ACES. Section 4 presents some additional examples and simulation results on real circuits using state transformation. Section 5 summarizes the paper.

2. State Dependency in Event Driven Explicit Simulation

Consider a simple 2RC circuit as shown in Fig. 4 This simple exam-

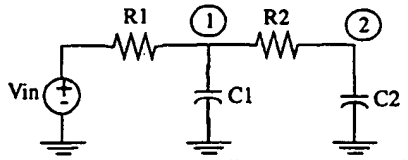

Fig. 4: *A simple 2RC example to illustrate the state dependency problem*

ple is used to review the basic ACES algorithm and to illustrate the problem of state dependence. The state equation of the circuit can be written as

$$\begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \end{bmatrix} = \begin{bmatrix} -\left(\frac{1}{R_1} + \frac{1}{R_2}\right) & \frac{1}{R_2} \\ \frac{1}{R_2} & -\frac{1}{R_2} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} V_{in} \qquad (1)$$

where the capacitor voltages have been chosen to be the state variables. Let V_{in} be a unit step function, $C_1 = C_2 = 1F$, $R_1 = 1\Omega$, and $R_2 = 10^{-6}\Omega$. Then the state equation can be written as

$$\begin{bmatrix} \dot{v}_1 \\ \dot{v}_2 \end{bmatrix} = \begin{bmatrix} -(1+10^6) & 10^6 \\ 10^6 & -10^6 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} V_{in} \qquad (2)$$

Assume that the circuit is initially at rest, i.e $v_1 = v_2 = 0$ at $t = 0$. Also assume a unit step input, i. e. $V_{in} = 1$ and $\dot{V}_{in} = 0$ at $t = 0^+$. Then, applying the ACES algorithm to compute the time to quiescence [2] for the two states yields $\Delta t_1 = -(\dot{v}_1(0))/(\ddot{v}_1(0)) = 10^{-6}$ and $\Delta t_2 = 0$, which implies that the input excitation has not reached the second node at this time point. In other words, one is interested only in non-zero time step at each time point. Hence, $\Delta t = \Delta t_1 = 10^{-6}$.

Continue the ACES algorithm to the next time point, $t_1 = t_0 + \Delta t = 10^{-6}$. Note that v_1 is now in quiescence and it should not dictate the computation of the next time step. Hence the time to quiescence for v_2 is given by

$$\hat{\Delta t} = -(\dot{v}_2(t_1))/(\ddot{v}_2(t_1)) = 1 + 10^{-6}.$$

Then, at the next time point, $t_2 = t_1 + \hat{\Delta t}$, both states will be quiescence and the simulation is completed. The result of this simple simulation for v_2 is shown in Fig. 5 together with the result obtained by AS/X, the IBM internal circuit simulator. Fig. 5 shows the result obtained by ACES with a voltage error tolerance $\varepsilon_v = 0.001$, where

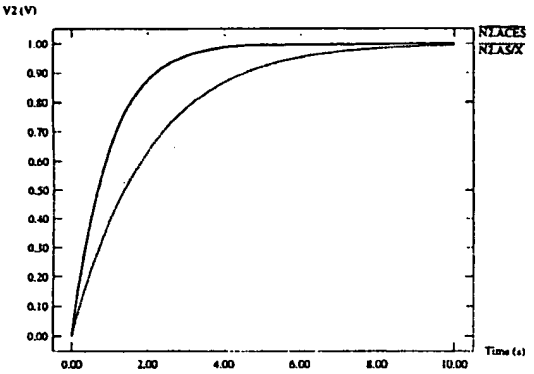

Fig. 5: *ACES simulation result (without state transformation) as compared to AS/X for the simple 2RC circuit shown in Fig. 4 with an error tolerance of 0.001.*

$\varepsilon_v = \frac{1}{2}\ddot{v}(\Delta t)^2$. Even with this fairly tight tolerance, the ACES result shows significant discrepancy as compared to the AS/X result. This problem is due to the fact that the two states are treated as independent and processed one at a time.

From the circuit point of view, when v_1 enters quiescence, ACES replaces C_1 by a zero valued current source, i. e. C_1 becomes an open circuit and the time constant of this new circuit is given by $\tau = (R_1 + R_2)C_2 = 1 + 10^{-6}$. This is exactly the time to quiescence or the time to steady state for this new circuit, assuming a first order approximation for v_2. However, this is not the correct time constant for the original circuit. Ignoring the effect of the small resistor R_2, the correct time constant is given by $\tau = R_1(C_1 + C_2) = 2$. This discrepancy illustrates the problem in assuming the independence of strongly coupled state variables.

In order to resolve this problem, an appropriate state space representation of the circuit must be chosen in such a way that the states of this space are truly independent. One obvious choice is the eigen-space of the circuit. Compute the eigen-decomposition of the system matrix as

$$A = \begin{bmatrix} -\left(\frac{1}{R_1} + \frac{1}{R_2}\right) & \frac{1}{R_2} \\ \frac{1}{R_2} & -\frac{1}{R_2} \end{bmatrix} = \begin{bmatrix} -(1+10^6) & 10^6 \\ 10^6 & -10^6 \end{bmatrix} = T\Lambda T^{-1} \qquad (3)$$

where

$$T = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}, \; T^{-1} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}, \text{and } \Lambda = \begin{bmatrix} -2(10^6) & 0 \\ 0 & -0.5 \end{bmatrix} \qquad (4)$$

Define the transformed state variables as $\begin{bmatrix} y_1 & y_2 \end{bmatrix}^T = T^{-1}\begin{bmatrix} v_1 & v_2 \end{bmatrix}^T$. The transformed state equation can then be written as

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \Lambda\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + T\begin{bmatrix} 1 \\ 0 \end{bmatrix} V_{in} = \Lambda\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix} V_{in} \qquad (5)$$

Now apply the ACES algorithm to this system. At $t = 0$, $y_1 = y_2 = 0$. From the state equations, it can be seen that $\dot{y}_1 \neq 0$, $\dot{y}_2 \neq 0$, $\ddot{y}_1 \neq 0$, and $\ddot{y}_2 \neq 0$. Hence the time step can be computed as

$$\Delta t = min(\Delta t_1, \Delta t_2) = min\left(-\frac{\dot{y}_1}{\ddot{y}_1}, -\frac{\dot{y}_2}{\ddot{y}_2}\right) = \frac{1}{2}(10^{-6})$$

At the next time point $t_1 = \Delta t = 0.5(10^{-6})$, $\dot{y}_1 = \ddot{y}_2 = 0$. Hence y_1 will stay in quiescence and the time to quiescence for y_2 can be computed as

$$\hat{\Delta t} = -\frac{\dot{y}_2}{\ddot{y}_2} = 0.5$$

In this case, the ACES algorithm captures the correct time constant of the original circuit. This simple example has clearly illustrated the need to select an appropriate state space in which the state variables are truly independent so that an event driven explicit algorithm such as ACES can be applied to provide accurate simulation results.

3. State Transformation for Event Driven Explicit Simulation

The eigen-space representation is an ideal space for event driven explicit simulation because the modes of the circuit are decoupled and can be processed sequentially. However, this approach is clearly impractical due to a number of reasons. First the computation of the eigen-decomposition is expensive. Moreover, once the eigen-decomposition is known, the solution can be computed directly without resorting to numerical integration. Avoiding the eigen-decomposition is exactly the goal of numerical integration algorithm. So the question is how to determine an appropriate state transformation. One possible approach is to perform an approximate or inexact eigen-computation. This alleviates the cost of a full eigen-computation. However, for non-linear circuits, an eigen-computation needs to be performed for every new linearization of the nonlinear equations. Another approach is to perform a partial eigen-computation. This is a basically a model reduction problem, especially for large linear circuits. This approach is a standard technique for reduced order modeling of large linear circuits in general nonlinear simulation of circuits with large linear subcircuits. The advantage of this approach is that it can prune out the non dominant eigen-states, such as the state due to the large eigenvalue ($2(10^6)$) in the 2RC example above. However, this approach is not practical for nonlinear circuits.

In general, there is a need to have a simple transformation matrix in order not to degrade the efficiency of event driven explicit simulation technique such as ACES. Note again that one key assumption of circuit level (not logic level or macromodel or block level) event driven simulation is that the states are independent or DC decoupled so that the excitation can be propagated from the input source to the various nodes in the circuit in an event driven manner. For example, the unit step excitation from the input source of the above 2RC circuit will propagate first to node 1, and then to node 2. However, when the two nodes are strongly DC coupled, the algorithm will yield erroneous results as discussed in the previous section. The key idea is to process the two states simultaneously. Note that the eigen-transformation described in the previous section transforms the input matrix

$[1\ 0]^T$ to $[1/\sqrt{2}\ 1/\sqrt{2}]^T$. In the original state space, the input directly affects only v_1. In the transformed state space, the input directly affects both y_1 and y_2 simultaneously. Hence the key idea in constructing a simple transformation matrix is to make sure that all the transformed states will be dependent on all the original states. In this way, any input that affects at least one state in the original space will affect all the states in the transformed space. In this way, the set of coupled transformed states can be processed simultaneously during the "electrical" event propagation. For the 2RC circuit example, one such simple transformation matrix is given as

$$T = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \tag{6}$$

This matrix looks like a simple scaling of the transformation matrix T^{-1} (the inverse eigenvector matrix) discussed in the previous section. In general, this is not the case. The transformation matrix in Eq. (6) is simply to make every transformed states dependent upon all the original state variables. In order to bring out the effect of the capacitor values, let 's choose $C_1 = 0.5$ and $C_2 = 2$. Also select the capacitor charges, denoted by q_1 and q_2 as the state variables in the original space, i. e. $[q_1\ q_2]^T = [C_1 v_1\ C_2 v_2]^T$. And the transformed state variables are defined as

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = T \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

An ACES simulation is then performed for this circuit. Note that the times to quiescence (an "electrical" event in ACES) are computed for the transformed state variables. The local truncation error control is performed for the original state variables (the node voltages) in this case. The waveform at node 2 as obtained by the ACES simulation with state transformation is shown in Fig. 6 together with the AS/X

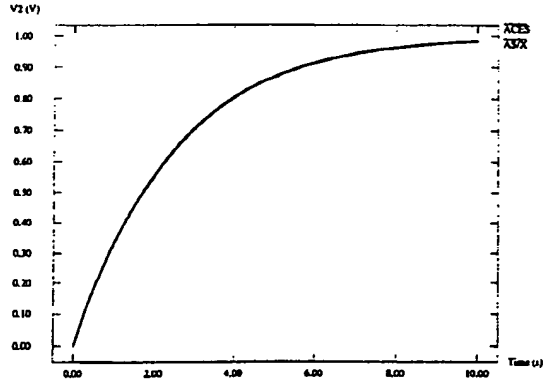

Fig. 6: *ACES simulation result (with state transformation) as compared to AS/X for the simple 2RC circuit shown in Fig. 4*

simulation result. Note that the state transformation helps ensure the accuracy of ACES simulation.

In general, if there are no coupling among the states, then the transformation matrix is simply an identity matrix. Suppose there are n coupled states, then the submatrix corresponding to these n states will be of the form

$$T_{nxn} = \begin{bmatrix} +1 & -1 & -1 & \dots & -1 \\ +1 & +1 & -1 & \dots & -1 \\ +1 & +1 & +1 & \dots & -1 \\ & & & \dots & \\ +1 & +1 & +1 & \dots & +1 \end{bmatrix}$$

For example, given a circuit with 7 states among which there are two sets of coupled states: {2,3,4} and {6,7}, then the transformation matrix for this circuit can be written as

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

This simple structure allows the transformation matrix to be constructed in a systematic manner during the processing of the circuit. For example, a simple two phase procedure to construct this type of transformation matrix can be described as follows. First initialize the transformation matrix T to be an identity matrix. Then in the first phase of processing the circuit, check if the terminals (which are assumed to be states for simplicity) or nodes of a given resistive element (or device) are coupled. If the nodes are coupled, then enter the appropriate nonzero entries, which corresponds to the indices of the nodes in the state vector, in the transformation matrix. The purpose of this phase is to record the coupling information in the matrix. In the second phase, the transformation matrix will be processed row by row to determine different groups of coupled states. Then the submatrices of the type described above for these groups of coupled states will be recorded in the transformation matrix.

The next question is how to incorporate state transformation into ACES. A straightforward approach is simply transform the original state equation into a new state equation and perform the analysis in the transformed space. However, it should be noted that the transformed space is used to compute the time to quiescence for the transformed states while the local truncation error is still performed in the original space. Therefore, it is simpler to impose the constitution relation or the quiescence condition for the transformed states in terms of the original states to solve for the necessary variables in the original space. Then the variables in the transformed space can be computed by applying the transformation matrix to the original states. These ideas will be illustrated below using a simple Modified Nodal Analysis (MNA) circuit formulation in the derivative space [2].

In order to simplify the presentation and to bring out the difference between the basic ACES algorithm and the modified algorithm to handle state transformation, assume that there are no inductors,

current sources, floating capacitors, and floating voltage sources. These circuit elements can be incorporated into the formulation in a straightforward manner once the basic algorithm has been established. Under these assumptions, during a quiescence period [2], the MNA formulation of the circuit in the derivative space can be written as

$$\begin{bmatrix} G_{cc} & G_{cv} & E_{cc} & 0 \\ G_{vc} & G_{vv} & 0 & E_v \\ K_c & 0 & E_{qc} & 0 \\ 0 & U_v & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{v}_c \\ \dot{v}_v \\ \ddot{q}_c \\ \ddot{q}_v \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ h_c \\ \dot{v}_v \end{bmatrix} \tag{7}$$

where G_{cc}, G_{cv}, G_{vc}, and G_{vv} are submatrices of the conductance matrix formed by resistive elements; E_{cc}, E_{qc}, E_v, and U_v are submatrices of 1's and 0's to pick out the appropriate capacitor/source voltages or currents; K_c and h_c represent the changeable portion of the MNA equation during the simulation depending upon whether the states are in quiescence or not. In general, the first two rows of Eq. (7) represents Kirchoff Current Law (KCL) equations at all voltage and capacitance nodes. The third row represents the Branch Constitutive Relations (BCR) and/or the quiescence conditions for the capacitors. The fourth row represents the BCRs for the voltage sources.

Now define the state transformation $y_c = Tq_c$. Initially when all the states are non-quiescent, the BCRs for the transformed states can be written as $\dot{y}_c = TC\dot{v}_c$. In circuit theoretic terms, the transformed states can be considered as voltage controlled current sources. In this case, the stamps in the MNA matrix of Eq.(7) can be set as follows: $K_c = TC$, $E_{qc} = 0$, and $h_c = \dot{y}_c$. Suppose at time t, the jth (transformed) state enters quiescence. Then $\ddot{y}_{cj}(t) = t_j^T \ddot{q}_c(t) = 0$ and the stamps in the MNA matrix can be modified as $k_{cj}^T = 0$, $e_{qc,j}^T = t_j^T$, and $h_{cj} = 0$, where t_j^T is the jth row of T, k_{cj}^T is the jth row of K_c, and $e_{qc,j}^T$ is the jth row of E_{qc}. In this case, the transformed state in quiescence can be treated as a zero valued current controlled current source. In this case, partition the vector of transformed states as

$$y_c = \begin{bmatrix} y_{cn} \\ y_{cs} \end{bmatrix} = \begin{bmatrix} T_n \\ T_s \end{bmatrix} q_c \tag{8}$$

where y_{cn} denotes the subvector of transformed states that are not in quiescence, and y_{cs} denotes the subvector of transformed states that are in quiescence; T_n and T_s are appropriate partitions of the transformation matrix T. Then the MNA equation as shown in Eq.(7) can be rewritten as

$$\begin{bmatrix} G_{cc} & G_{cv} & E_{cc} & 0 \\ G_{vc} & G_{vv} & 0 & E_v \\ T_n C & 0 & 0 & 0 \\ 0 & 0 & T_s & 0 \\ 0 & U_v & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{v}_c \\ \dot{v}_v \\ \ddot{q}_c \\ \ddot{q}_v \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \dot{y}_c \\ 0 \\ \dot{v}_v \end{bmatrix} \tag{9}$$

where the third row of Eq.(9) describes the non-quiescent transformed states, and the fourth row describes the quiescent transformed states. The modified ACES algorithm with state transformation is summarized in the pseudocode below.

Algorithm with state transformation:

1. Update sources

2. Update original state variables (for all states):

$\dot{q}_c(t\text{-}) = \dot{q}_c(t-\Delta t) + \ddot{q}_c(t-\Delta t)\Delta t$ *and*

$v_c(t\text{-}) = v_c(t-\Delta t) + \dot{v}_c(t-\Delta t)\Delta t$

2b. Update the transformed states:

$\dot{y}_c(t\text{-}) = T\dot{q}_c(t\text{-})$ for quiescent states, and $\dot{y}_c(t\text{-}) = 0$ for quiescent states

3. Check quiescence conditions for y_c

4. Setup the MNA matrix and solve for $\dot{v}_c(t+)$, $\ddot{q}_c(t)$, *and* $\ddot{q}_v(t)$.

5. Compute $\dot{q}_c(t+) = C\dot{v}_c(t+)$ *and* $\ddot{v}_c(t) = C^{-1}\ddot{q}_c(t)$ *for all states*

5b. Compute $\dot{y}_c(t+) = T\dot{q}_c(t+)$ *and* $\ddot{y}_c(t) = T\ddot{q}_c(t)$ *for non-quiescent states*

6. Compute time to quiescence for non-quiescent states, $t_{qs} = -[\dot{y}_c(t)/(\ddot{y}_c(t))]$

7. Set $\Delta t = min(t_{qs})$

In the above algorithm, the quiescence conditions are checked for the transformed states, not the original states. The advantage of this algorithm is that there is no need to explicitly construct the transformed space. The solution procedure is still performed in the original space with minimal modification of the matrix equation to impose the BCRs and/or the quiescence conditions for the transformed states. In the next section, some examples will be presented to assess the overhead of the modified ACES algorithm and its accuracy as compared with the basic ACES algorithm.

4. Examples

The first example is a simple nonlinear circuit consisting of a driver driving four buffers. Each connection between the driver and a given buffer is an 8 RC interconnect model. The schematic of the circuit is shown in Fig. 7. This example is to illustrate the complication due to circuit partitioning which may couple a port to an internal state of a given partition. This situation arises when parasitic resistances are added to the terminals of a MOSFET model. For example, there are five partitions P1 to P5 in the circuit shown in Fig. 7. Ports numbered 1 to 3 will be coupled to the internal states of partition P1 if the resistor at the end of the lines T1 to T3 are shorted. In a standard ACES algorithm, a partition is processed only if a port attached to the partition has an event. A port event is initiated when the volt-

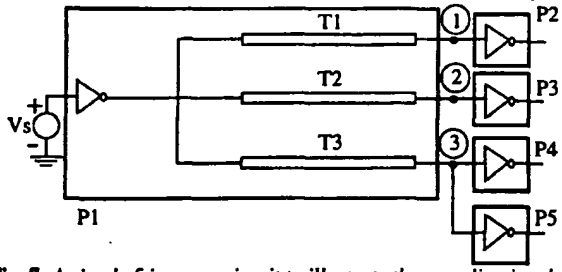

Fig. 7: *A simple 5 inverter circuit to illustrate the complication due to circuit partitioning when using state transformation*

age slope at the port has an appreciable change. Otherwise, the partition is not processed even if the port information is updated when another partition connected to the same port has an event. For example, assume partition P2 has an event at a given time point. In processing partition P2, the voltage and the slope at port 1 will be updated. If the slope has a significant change (above a given tolerance), then partition P1 will be processed at the same event time. Otherwise, P1 will not be processed. However, in the case port 1 is coupled to an internal state of P1, P1 will have to be processed regardless of whether P1 has a significant change in voltage slope or not. This processing is necessary to take the coupling into account when computing all the necessary information for the next event. A sample run is performed for this circuit with the first, last, and two middle resistors of the lines are shorted to produce different modes of coupling among the internal states and the ports of P1. In this case, the ACES simulation without state transformation requires 309718 events and 34.36 CPU seconds. The ACES simulation with state transformation requires only 3822 events and 2.77 CPU seconds. The ACES simulation results are compared with AS/X simulations and shown in Fig. 8 at the output of partition P2.

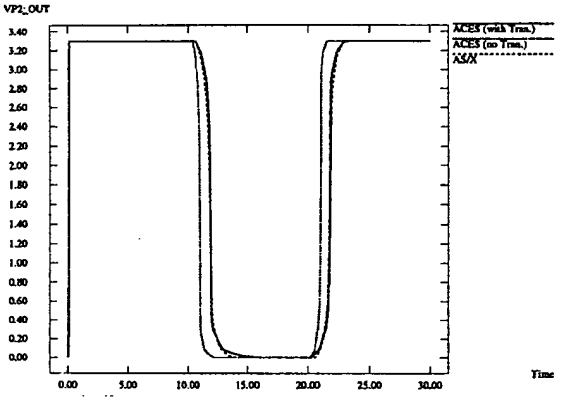

Fig. 8: *ACES simulation results (with and without state transformation) as compared to AS/X at the output of partition P2 as shown in Fig. 7.*

The last example is an industrial design of a 4 bit ALU. The circuit consists of 282 MOSFETs and 156 nodes. Without parasitic resistances added to the MOSFET models, the ACES simulation requires only 7298 events and 1.17 CPU seconds while the AS/X simulation requires 50 CPU seconds. With a number of parasitic resistances added to the circuit, the AS/X simulation requires about the same 50 CPU seconds. In this case, the ACES simulation with state transformation requires 17640 events and 5.73 CPU seconds. The waveforms at an output node of the circuit as obtained by ACES simulation with and without parasitic resistances and by AS/X are shown in Fig. 9. Note that the difference between the ACES waveforms is due to the state transformation. In the case of no parasitic resistances (and no transformation), the time to quiescence is computed for the original state. In the case of state transformation, the time to quiescence is computed for the transformed states.

5.Summary

This paper has identified the dependency of the states in the state space description of a given circuit as the cause for some numerical

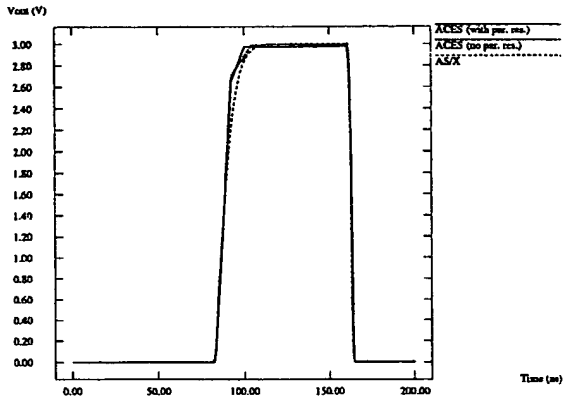

Fig. 9: *ACES simulation results (with and without parasitic resistances) as compared to AS/X at the output of an industrial 4 bit ALU circuit.*

problems in event driven explicit simulation algorithms such as ACES. State transformation has also been proposed to address this problem. A novel algorithm has been developed to derive systematically a simple state transformation and to include such a transformation in an efficient manner. A number of examples have been included to illustrate the concept and the complication in using state transformation as well as to demonstrate its effectiveness.

References

[1] J. T. J. Van Eijndhoven, A Piecewise Linear Simulator for Large Scale Integrated Circuits, Ph.D Thesis, Eindhoven Institute of Technology, 1984.

[2] A. Devgan and R. A. Rohrer, "Adaptively Controlled Explicit Simulation," *IEEE Trans. Computer-Aided Design*, vol. 13, pp.746-762, June 1994.

[3] Y.H. Shih and S. M. Kang, "ILLIADS: A Fast Timing and Reliability Simulator for Digital MOS Circuits," *IEEE Trans. Computer-Aided Design*, pp. 1387-1402, Sept. 1993.

[4] C. Visweswariah and R. A. Rohrer, "Piecewise Approximate Circuit Simulation," *IEEE Trans. Computer-Aided Design*, pp. 861-870, July 1991.

[5] A. D. Stein, T. V. Nguyen, B. George, and R. A. Rohrer, "ADAPTS: A Digital Transient Simulation Strategy for Integrated Circuits," *Proc. DAC*, June 1991.

[6] S. Lin, M. M. Sadowska, and E. S. Kuh, "Stepwise Equivalent Circuit Simulation Technique," *IEEE Trans. Computer-Aided Design*, pp. 672-683, May 1993.

[7] R. Kao and M. Horowitz, "Timing Analysis using Piecewise Linear RSIM," *IEEE Trans. Computer-Aided Design*, pp. 1498-1512, Dec. 1994.

Your search matched **40** of **1024576** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enter new one in the text box.

(event <near/5> driven) <and> (state) <and> (circuit Search

Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

---

16 **An innovative procedure for reliability assessment of power electr equipped systems: a real case study**
Fazio, V.; Savio, S.; Firpo, P.;
Industrial Electronics, 2000. ISIE 2000. Proceedings of the 2000 IEEE Interna Symposium on , Volume: 2 , 4-8 Dec. 2000
Pages:511 - 516 vol.2

[Abstract] [PDF Full-Text (656 KB)] IEEE CNF

---

17 **Controlling state explosion in static simulation by selective comp**
Chakrabarti, P.P.; Dasgupta, P.; Das, P.P.; Roy, A.; Lahiri, S.; Bose, M.;
VLSI Design, 1999. Proceedings. Twelfth International Conference On , 7-10 1999
Pages:226 - 231

[Abstract] [PDF Full-Text (80 KB)] IEEE CNF

---

18 **Parallel logic simulation on a network of workstations using PVM**
Kormicki, M.; Mahmood, A.; Carlson, B.S.;
Parallel and Distributed Processing, 1996. Eighth IEEE Symposium on , 23-26 1996
Pages:2 - 9

[Abstract] [PDF Full-Text (728 KB)] IEEE CNF

---

19 **Cycle-based timing simulati ns using event-streams**
Kei-Yong Khoo; Willson, A.N., Jr.;
Computer Design: VLSI in Computers and Processors, 1996. ICCD '96. Proceedings., 1996 IEEE International Conference on , 7-9 Oct. 1996
Pages:460 - 465

[Abstract] [PDF Full-Text (672 KB)] IEEE CNF

20 **A multiple-d minance switch-level m del f r simulati n f sh rt fa**
Dahlgren, P.;
Computer-Aided Design, 1995. ICCAD-95. Digest of Technical Papers., 1995 IEEE/ACM International Conference on , 5-9 Nov. 1995
Pages:674 - 680

[Abstract] [PDF Full-Text (624 KB)] IEEE CNF

21 **Controlling change propagation and project policies in IC design**
Mathys, Y.; Morgan, M.; Soudagar, S.;
European Design and Test Conference, 1995. ED&TC 1995, Proceedings. , 6-
March 1995
Pages:274 - 279

[Abstract] [PDF Full-Text (464 KB)] IEEE CNF

22 **Variable accuracy device modeling for event-driven circuit simulati**
Michaels, K.W.; Strojwas, A.J.;
European Design and Test Conference, 1994. EDAC, The European Conferenc Design Automation. ETC European Test Conference. EUROASIC, The Europea Event in ASIC Design, Proceedings. , 28 Feb.-3 March 1994
Pages:557 - 561

[Abstract] [PDF Full-Text (376 KB)] IEEE CNF

23 **Macromodeling CMOS circuits for event driven simulation**
Trotter, J.D.; Saripella, S.; Pidugo, N.; Ledlow, D.L.; Kapoor, D.;
ASIC Conference and Exhibit, 1994. Proceedings., Seventh Annual IEEE International , 19-23 Sept. 1994
Pages:174 - 177

[Abstract] [PDF Full-Text (272 KB)] IEEE CNF

24 **A logic simulation engine based on a modified data flow architectu**
Mahmood, A.; Baker, W.I.; Herath, J.; Jayasumana, A.;
Computer-Aided Design, 1992. ICCAD-92. Digest of Technical Papers., 1992 IEEE/ACM International Conference on , 8-12 Nov. 1992
Pages:377 - 380

[Abstract] [PDF Full-Text (320 KB)] IEEE CNF

25 **Comparisons and analysis of massively parallel SIMD architectures parallel logic simulation**
Choi, E.M.; Chung, M.J.; Chung, Y.;
Parallel Processing Symposium, 1992. Proceedings., Sixth International , 23-
March 1992
Pages:971 - 974

[Abstract] [PDF Full-Text (388 KB)] IEEE CNF

26 **Incremental-in-time alg rithm f r digital simulati n**
Choi, K.; Hwang, S.Y.; Blank, T.;
Design Automation Conference, 1988. Proceedings., 25th ACM/IEEE , 12-15 J 1988
Pages:501 - 505

[Abstract] [PDF Full-Text (380 KB)] **IEEE CNF**

---

27 **M^3-a multilevel mixed-mode mixed D/A simulator**
Chadha, R.; Chen, C.-F.;
Computer-Aided Design, 1988. ICCAD-88. Digest of Technical Papers., IEEE International Conference on , 7-10 Nov. 1988
Pages:258 - 261

[Abstract] [PDF Full-Text (312 KB)] **IEEE CNF**

---

28 **Time domain current waveform simulation of CMOS circuits**
An-Chang Deng; Yan-Chyuan Shiau; Loh, K.-H.;
Computer-Aided Design, 1988. ICCAD-88. Digest of Technical Papers., IEEE International Conference on , 7-10 Nov. 1988
Pages:208 - 211

[Abstract] [PDF Full-Text (292 KB)] **IEEE CNF**

---

29 **A New Approach to Hierarchical and Statistical Timing Simulations**
Benkoski, J.; Strojwas, A.J.;
Computer-Aided Design of Integrated Circuits and Systems, IEEE Transaction on , Volume: 6 , Issue: 6 , November 1987
Pages:1039 - 1052

[Abstract] [PDF Full-Text (2376 KB)] **IEEE JNL**

---

30 **Switch-Electrical Segmented Waveform Relaxation for Digital MOS and Its Acceleration on Parallel Computers**
Dumlugol, D.; Odent, P.; Cockx, J.P.; De Man, H.J.;
Computer-Aided Design of Integrated Circuits and Systems, IEEE Transaction on , Volume: 6 , Issue: 6 , November 1987
Pages:992 - 1005

[Abstract] [PDF Full-Text (1976 KB)] **IEEE JNL**

Prev 1 2 3 Next

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search | Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting | No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top

Terms used **event driven state driven circuit** Found **1,874** of **131,734**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

1 Structural active object systems for simulation

Toshimi Minoura, Shirish S. Pargaonkar, Kurt Rehfuss

October 1993 **ACM SIGPLAN Notices , Proceedings of the eighth annual conference on Object-oriented programming systems, languages, and applications**, Volume 28 Issue 10

Full text available: pdf(1.83 MB) Additional Information: full citation, references, citings, index terms

2 Event driven adaptively controlled explicit simulation of integrated circuits

Anirudh Devgan, Ronald A. Rohrer

November 1993 **Proceedings of the 1993 IEEE/ACM international conference on Computer-aided design**

Full text available: pdf(478.36 KB) Additional Information: full citation, references, citings

3 LECSIM: a levelized event driven compiled logic simulation

Zhicheng Wang, Peter M. Maurer

January 1991 **Conference proceedings on 27th ACM/IEEE design automation conference**

Full text available: pdf(783.84 KB) Additional Information: full citation, abstract, references, citings, index terms

LECSIM is a highly efficient logic simulator which integrates the advantages of event driven interpretive simulation and levelized compiled simulation. Two techniques contribute to the high efficiency. First it employs the zero-delay simulation model with levelized event scheduling to eliminate most unnecessary evaluations. Second, it compiles the central event scheduler into simple local scheduling segments which reduces the overhead of event scheduling. Experimental results show that LECS ...

4 A case against event-driven simulation for digital system design

Glenn Jennings

April 1991 **Proceedings of the 24th annual symposium on Simulati n**

Full text available: pdf(699.67 KB) Additional Information: full citation, references, citings, index terms

5 State transformation in event driven explicit simulation

Tuyen V. Nguyen, Anirudh Devgan

November 1997 **Proceedings f the 1997 IEEE/ACM international c nference n C mputer-aided design**

Full text available: pdf(68.29 KB) Additional Information: full citation, abstract, references, index terms

Publisher Site

This paper presents a general method for incorporating state transformation in event driven explicit simulation. One inherent assumption in this type of simulation algorithm is the state independence, which allows the algorithm to process the states independently in an event driven manner at the transistor level. Numerical problems arise when an inappropriate state representation of the circuit, in which the states are not truly independent, is chosen. In principle, any similarity transformation ...

Keywords: circuit analysis computing, event driven explicit simulation, similarity transformation, simulation algorithm, state equation, state representation, state transformation

6 A precise event-driven circuit simulator based on predicted fan-in voltages

H. Fujisawa, F. Kawafuji, T. Kitaura, T. Kage

March 1995 **Proceedings of the 1995 European conference on Design and Test**

Full text available: pdf(92.87 KB) Additional Information: full citation, abstract

Publisher Site

We propose a new event-driven circuit simulation method for MOS transistor circuits. This method is based on predicted and revised voltages of nodes, and is highly accurate. Furthermore this method can use an effective block selection function (EBSF) which allows faster simulation with the same accuracy. In industrial circuits, actually our method achieved accuracy equal to or higher than that of a SPICE-like simulator at 3 to 5 times the speed without EBSF, or 11 to 22 times the speed with EBSF ...

Keywords: MOS integrated circuits, MOS transistor circuits, MOSFET circuits, circuit analysis computing, effective block selection function, event-driven circuit simulator, predicted fan-in voltages

7 Incremental event-driven simulation of digital FET circuits

Chandramouli Visweswariah, Jalal A. Wehbeh

July 1993 **Proceedings of the 30th international on Design automation conference**

Full text available: pdf(445.39 KB) Additional Information: full citation, references, citings, index terms

8 Parallel-and-vector implementation of the event-driven logic simulation algorithm on the Cray Y-MP supercomputer

A. Bataineh, F. Özgüner

December 1992 **Proceedings of the 1992 ACM/IEEE conference on Supercomputing**

Full text available: pdf(926.62 KB) Additional Information: full citation, references, index terms

9 Efficient circuit partitioning algorithms for parallel logic simulation

S. Patil, P. Banerjee, C. Polychronopoulos

August 1989 **Pr ceedings of the 1989 ACM/IEEE conference n Supercomputing**

Full text available: pdf(959.59 KB) Additional Information: full citation, abstract, references, citings, index terms

General purpose parallel processing machines are increasingly being used to speed up a variety of VLSI CAD applications. This paper addresses logic simulation on parallel machines by exploiting the concurrency in the circuit being simulated (called data parallelism) as opposed to exploiting parallelism inherent in the simulation algorithm itself (called functional parallelism). The most crucial step in obtaining the maximum parallelism using data parallelis ...

10 Parallel compiled event driven VHDL simulation

V. Krishnaswamy, P. Banerjee

July 1998 **Proceedings of the 12th international conference on Supercomputing**

Full text available: pdf(964.02 KB) Additional Information: full citation, references, index terms

11 Power estimation tool for sub-micron CMOS VLSI circuits

F. Rouatbi, B. Haroun, A. J. Al-Khalili

November 1992 **Proceedings of the 1992 IEEE/ACM international conference on Computer-aided design**

Full text available: pdf(565.03 KB) Additional Information: full citation, references, citings, index terms

12 A Parallel and Accelerated Circuit Simulator with Precise Accuracy

Peter M. Lee, Shinji Ito, Takeaki Hashimoto, Tomomasa Touma, Hitachi ULSI Systems Co., Junji Sato, Goichi Yokomizo, Semiconductor, Ic, Hitachi, Ltd

January 2002 **Proceedings of the 2002 conference on Asia South Pacific design automation/VLSI Design**

Full text available: pdf(128.60 KB) Publisher Site Additional Information: full citation, abstract

We have developed a hi ghly parallel and accelerated circuit simulator which produces precise results for large scale simulation. We incorporated multithreading in both the model and matrix calculations to achieve not only a factor of 10 acceleration compared to the defacto standard circuit simulator used worldwide, but also equal or exceed the performance of timing-based event -driven simulators with the accuracy which matches that of SPICE-based circuit simulation. For example, a 89K element D ...

13 Performance evaluation of an event-driven logic simulation machine

F. Hirose

July 1992 **Proceedings of the 29th ACM/IEEE conference on Design automation conference**

Full text available: pdf(345.73 KB) Additional Information: full citation, references, index terms

14 A fast signature simulation tool for built-in self-testing circuits

S. B. Tan, K. Totton, K. Baker, P. Varma, R. Porter

October 1987 **24th ACM/IEEE c nference pr ceedings n Design automati n c nference**

Full text available: pdf(896.99 KB) Additional Information: full citation, abstract, references, index terms

This paper describes a Fast Signature Simulator (FSS) tool for Built-In Self-Testing (BIST) circuits. The FSS consists of a simulator generator and a compiled code simulator. The

simulator generator comprises a controlling program called the EXECUTIVE and translation software called SIM-GEN. SIM-GEN accepts a Hardware Description Language (HDL) representation of the circuit-under-test as its input and produces C code simulation modules comprising Boolean relations that represent the structu ...

15 An evaluation of the Chandy-Misra-Bryant algorithm for digital logic simulation

Larry Soulé, Anoop Gupta

October 1991 **ACM Transactions on Modeling and Computer Simulation (TOMACS)**, Volume 1 Issue 4

Full text available: pdf(2.64 MB) Additional Information: full citation, abstract, references, citings, index terms

We explore the suitability of the Chandy-Misra-Bryant (CMB) algorithm for the domain of digital logic simulation. Our evaluation is based on results for six realistic benchmark circuits, one of them being the R6000 microprocessor form MIPS. A quantitative evaluation of the concurrency exhibited by the CMB algorithm shows that an average of 42-196 element activations can be evaluated in parallel if arbitrarily many processors are available. One major factor limiting the parallel performance ...

16 A framework for scheduling multi-rate circuit simulation

A. P.-C. Ng, V. Visvanathan

June 1989 **Proceedings of the 1989 26th ACM/IEEE conference on Design automation conference**

Full text available: pdf(745.68 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents a theoretical framework for scheduling of subcircuit simulation in a multirate simulation environment. We show that event-driven simulation, selective-trace, and latency are subsumed by this framework. We assume that the circuit to be simulated is partitioned into subcircuits and that the dependency relations can be expressed as a directed acyclic graph. Each subcircuit predicts its own stepsize, and we assume ...

17 Simulation I: An event-driven transient simulation algorithm for MOS and bipolar circuits

D. Patrick, C. Lyden

March 1990 **Proceedings of the conference on European design automation**

Full text available: pdf(320.34 KB) Additional Information: full citation, abstract, references

SUGAR is a program for fast transient simulation of MOS circuits [1]. This paper describes the extension to the algorithm to handle bipolar and BiCMOS circuits. The authors believe that SUGAR is the first event driven circuit simulator which successfully simulates bipolar circuits. The algorithm uses both dynamic and static circuit partitioning to identify closely coupled nodes, and performs simultaneous equation solution on the coupled nodes. A number of other extensions have been made to the a ...

18 Parallel logic simulation of VLSI systems

Mary L. Bailey, Jack V. Briner, Roger D. Chamberlain

September 1994 **ACM Computing Surveys (CSUR)**, Volume 26 Issue 3

Full text available: pdf(3.74 MB) Additional Information: full citation, abstract, references, citings, index terms

Fast, efficient logic simulators are an essential tool in modern VLSI system design. Logic simulation is used extensively for design verification prior to fabrication, and as VLSI systems grow in size, the execution time required by simulation is becoming more and more significant. Faster logic simulators will have an appreciable economic impact, speeding time to market while ensuring more thorough system design testing. One approach to this problem is to utilize parallel processing, taking ...

Keywords: circuit structure, parallel architecture, parallelism, partitioning, synchronization algorithm, timing granularity

19 A predictive system shutdown method for energy saving of event-driven computation

Chi-Hong Hwang, Allen C.-H. Wu

April 2000 **ACM Transactions on Design Automation of Electronic Systems (TODAES)**, Volume 5 Issue 2

Full text available: pdf(121.23 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents a system-level power management technique for energy savings of event-driven application. We present a new predictive system-shutdown method to exploit sleep mode operations for energy saving. We use an exponential-average approach to predict the upcoming idle period. We introduce two mechanisms, prediction-miss correction and prewake-up, to improve the hit ratio and to reduce the delay overhead. Experiments on four different event-driven applications show that our propo ...

Keywords: event-drive, power management, predictive, sleep mode, system shutdown

20 Incremental circuit simulation using waveform relaxation

Y.-C. Ju, R. A. Saleh

July 1992 **Proceedings of the 29th ACM/IEEE conference on Design automation conference**

Full text available: pdf(456.22 KB) Additional Information: full citation, references, index terms

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Google Web Images Groups News Froogle[New!] more »
"event driven" "state driven" Search Advanced Search Preferences

W b Results 1 - 10 of about 178 for "event driven" "state driven". (0.18 seconds)

Event Driven Programming
... Once described this way (CeeBias?): "A **state-driven** program has an outer switch(on state), with inner switches(on event). An **event-driven** program has an outer ...
c2.com/cgi/wiki?EventDrivenProgramming - 5k - Apr 18, 2004 - Cached - Similar pages

> **Quick Change**
> ... Once described this way (CeeBias): 27a31,35 "A **state-driven** program has an outer switch(on state), with inner switches(on event). An **event-driven** program has ...
> c2.com/cgi/quickDiff?EventDrivenProgramming - 2k - Cached - Similar pages

Weblogs Forum - Why Are Event Driven APIs Difficult?
... **driven** APIs are more difficult because the **event-driven** environment is more complex. Will generators help you in your desire to simplfy **state-driven** machines? ...
www.artima.com/ forums/flat.jsp?forum=106&thread=8343 - 26k - Apr 17, 2004 - Cached - Similar pages

Background: Event v. State Driven Design
Event Driven - based on interesting algorithm events: Difficult/Time intensive; Excellent abstraction. **State Driven** - based on the status of data: ...
www.cs.hope.edu/~sumner/jsave/ ccsc2003/04EventState.html - 2k - Cached - Similar pages

Finite State Machines (Spring 1998)
... approach where inputs/outputs (stimuli/responses) are represented by events and each finite state machine is **event-driven**, rather than **state-driven** as shown in ...
www.csc.uvic.ca/~mcheng/460/notes/fsm2.html - 7k - Cached - Similar pages

[aosd-discuss] Re: Aspects and the Event-Driven Model
... and the **Event-Driven** Model; Next message: [aosd-discuss] Aspects and the **Event-Driven** Model; ... of rule based systems is that they are data/**state driven** and the ...
server2.hostvalu.com/pipermail/discuss_aosd.net/ 2003-August/000608.html - 9k - Cached - Similar pages

> **[aosd-discuss] Re: Aspects and the Event-Driven Model**
> ... at aosd.net Subject: [aosd-discuss] Re: Aspects and the **Event-Driven** Model How ... the strengths of rule based systems is that they are data/**state driven** and the ...
> server2.hostvalu.com/pipermail/discuss_aosd.net/ 2003-August/000609.html - 12k - Cached - Similar pages
> [More results from server2.hostvalu.com]

[PDF] **Event-Driven Computing Projects for Software Engineering Education**
File Format: PDF/Adobe Acrobat - View as HTML
... to **state-driven** models of dynamic systems. However, our observation is that students have not readily grasped the concepts of dynamic, **event-driven** systems ...
www.cs.missouri.edu/~skubic/Papers/skubicASEE2002.pdf - Similar pages

[PDF] **The PC Based Control System of the NAC**
File Format: PDF/Adobe Acrobat - View as HTML
... All operator initiated traffic is **event driv n** providing a quick response. The actual value readback is partly **state driven**, partly **event driven**. ...
www.aps.anl.gov/conferences/icalepcs/ 97/paper97/p016.pdf - Similar pages

OMG Test Special Interest Group - Austin
... Bob then stated that users must make sure they test with action driven, data driven, logic driven, **event driven**, and **stat driven** approaches. ...
testsig.omg.org/austin.htm - 16k - Cached - Similar pages

Gooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

"event driven" "state driven" Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Solutions - Business Solutions - About Google

Google Web Images Groups News Froogle[New!] more »
"event driven" "state driven" circuit simulati Search Advanced Search Preferences

Web Results **1 - 8** of **8** for **"event driven" "state driven" circuit simulation**. (**0.17** seconds)

Tip: Try removing quotes from your search to get more results.

[PDF] **Event-Driven** Computing Projects for Software Engineering Education
File Format: PDF/Adobe Acrobat - View as HTML
... in the past, introduced students to **state-driven** models of ... a testbed for exploring **event-driven** systems and ... The printed **circuit** boards were fabricated by MU ...
www.cs.missouri.edu/~skubic/Papers/skubicASEE2002.pdf - Similar pages

[PDF] Session F3C TEACHING REAL-TIME CONCEPTS IN AN UNDERGRADUATE ...
File Format: PDF/Adobe Acrobat - View as HTML
... combinational and sequential logic **circuit** design and ... The notion of "**event driven**" processes is emphasized ... Other structures such as **state-driven**, round robin ...
fie.engrng.pitt.edu/fie2003/papers/1383.pdf - Similar pages

From bruce@hpcea.ce.hp.com Fri Aug 14 15:11:26 1992 Received: from ...
... object paradigm leads to an **event-driven** model of ... be able to be described as **state driven** - each state ... one could develop a digital **circuit simulation** system as ...
wuarchive.wustl.edu/languages/smalltalk/ patterns/arch-handbook91 - 47k - Cached - Similar pages

Sponsored Links

Party Finder
The Premier Guide To Circuit Events Worldwide
www.partyfinder.com

Circuit Simulation
Analog, digital, mixed & symbolic circuit simulator. Download Now!
www.tina.com

Spice **Circuit Simulation**
Analog **circuit** simulator with schematic capture. Free download!
www.5Spice.com

Fast **Circuit** Simulator
Industrial-Proven MSIM & Turbo-MSIM Accurate, Big Capacity, Great Value
www.LegendDesign.com

A **Circuit Simulation** Tool
Visualize your PSpice output. See when elements switch states.
MoHAT.com

See your message here...

[PDF] CAN Microcomputers Catalog
File Format: PDF/Adobe Acrobat
... ISO TC22 SC3 WGI TF6) conformance testing •Design **simulation** Flash Optimized ... With CAN outlines suited for **event-driven** or **state-driven** environments our ...
www.m16c.de/PDF/Catalogs/can_Catalog.pdf - Similar pages

[PDF] Revised Preliminary Version 4K420 Lecture Notes Embedded Systems ...
File Format: PDF/Adobe Acrobat - View as HTML
... to present a framework for **simulation**-based design ... of them are adopted in **event-driven** real-time ... Control (SBC) Electronic controllers on **circuit** boards first ...
se.wtb.tue.nl/documentation/lecnotes/es/eslnotes.pdf - Similar pages

[PDF] 65/290/DC INTERNATIONAL ELECTROTECHNICAL COMMISSION TC65 ...
File Format: PDF/Adobe Acrobat - View as HTML
Page 1. I 65/290/DC 2002-03-29 INTERNATIONAL ELECTROTECHNICAL COMMISSION TC65: INDUSTRIAL PROCESS MEASUREMENT AND CONTROL Circulation ...
www.holobloc.com/stds/iec/tc65wg6/ liaison/dptf/65_290e_DC.pdf - Similar pages

[PS] Control Using Logic-Based Switching AS Morse

File Format: Adobe PostScript - View as Text
... accomplishes this is a logical **circuit** which carries ... of linear controllers, and an **event driven** switching logic ... synthesis of such a **state driven** switching logic ...

entity.eng.yale.edu/controls/1998/a.ps - Similar pages

[PDF] OneChip: An FPGA Processor With Reconfigurable Logic Ralph D. ...
File Format: PDF/Adobe Acrobat - View as HTML
... 3-2 3.1.2 Benchmark **Circuit** Characteristics . ... with the investigations outlined above, several CAD tools for FPGA design implementation and **simulation** were used ...
dspwork.com/ links/fpga/thesis-An%20FPGA%20Processor%20With%20Reconfigurable%20Logic.pdf - Supplemental Result - Similar pages

Web

Results **1** - **10** of about **905** for **"two types" "circuit simulation"**. (**0.28** seconds)

[PDF] Mixed-Signal Multi-Level **Circuit Simulation**: An Implicit Mixed ...
File Format: PDF/Adobe Acrobat
... levels of abstraction Challenge • Not in simulating **two types** of technologies ... Mixed-Signal Multi-Level **Circuit Simulation** An Implicit Mixed-Mode Solution Mixed ...
www.ntu.edu.sg/eee/eee6/LectureNotes/Pub/COE-2.pdf - Similar pages

[PDF] Phys 315/519 Transistors Part 1 Recall: **two types** of charge ...
File Format: PDF/Adobe Acrobat - View as HTML
Phys 315/519 Transistors Part 1 Recall: • **two types** of charge carriers in semiconductors: electrons & holes • **two types** of doped semiconductors: n-type ...
galileo.phys.virginia.edu/~rjh2j/ phys315/Transistors.pdf - Similar pages

[PDF] A Simulation Study of Simultaneous Switching Noise
File Format: PDF/Adobe Acrobat - View as HTML
... SSN is studied under different simulation conditions for **two types** of packages ... are used for simulations within a complete **circuit simulation** environment such as ...
www.sigrity.com/papers/ECTC2001/ECTC_LI1.pdf - Similar pages

SPICE Simulation: Low-Capacitance TVS Devices for High-Frequency ...
... Unlike most pn junction components, these **two types** of diodes have unique SPICE parameters that are not normally included in **circuit simulation** software. ...
www.ce-mag.com/archive/01/11/Hutchin.html - 37k - Cached - Similar pages

Section Two: Chapter Four
... There are **two types** of macrocells, hard and soft ... time, enable and disable times provided with simulation models, through either **circuit simulation** or parameter ...
nppp.jpl.nasa.gov/asic/Sect.2.4.html - 12k - Cached - Similar pages

Introduction
... SPICE is a general-purpose **circuit simulation** program for nonlinear dc, nonlinear ... **Two types** of transfer functions are allowed : one of the form (output voltage ...
bwrc.eecs.berkeley.edu/Classes/IcBook/ SPICE/UserGuide/overview.html - 14k - Cached - Similar pages

[PS] MOSFET Modeling and SPICE: Are you an educated model consumer?
File Format: Adobe PostScript - View as Text
... **circuit simulation**. ffl Require FET models which are. ... 7. The Formalism of the SPICE FET Models ffl Models contain **two typ s** of parameters. - process parameters. ...
www.sover.net/~dfoty/cicc1.ps - Similar pages

[PPT] Converting A 64b PowerPC Processor from CMOS Bulk to SOI ...
File Format: Microsoft Powerpoint 97 - View as HTML

... Design conditions set for **tw types** of analysis. ... The static timing models were compared against accurate **circuit simulation** using body initialization. ...
www.sigda.org/Archives/ProceedingArchives/Dac/Dac99/papers/1999/dac99/slides/dp49_03.ppt - Similar pages

[PDF] **Circuit Simulation**
File Format: PDF/Adobe Acrobat
Page 1. **Circuit Simulation** May 2003 Page 2. ... 1-2 Common **Circuit Simulation** Methods ....
eesof.tm.agilent.com/docs/rfdedoc2003A/ pdf/rfdecktsim.pdf - Similar pages

Silicon-On-Insulator Technology Bumps Up SoC Performance
... There are **two types** of SOI devices: fully depleted and partially depleted. ... Thus, the netlists used for SOI **circuit simulation** typically differ from those used ...
www.elecdesign.com/ Articles/Index.cfm?ArticleID=3402&pg=2 - Similar pages

Gooooooooogle ▶
Result Page: 1 2 3 4 5 6 7 8 9 10 **Next**

"two types" "circuit simulation" Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Solutions - Business Solutions - About Google

Google Web Images Groups News Froogle[New!] more »

"logic simulation" "transition simulation" Search Advanced Search Preferences

Web Results **1** - **4** of about **6** for **"logic simulation" "transiti n simulation"**. (**0.24** seconds)

Tip: Try removing quotes from your search to get more results.

A methodology for hardware verification based on **logic simulation**
... E. Bryant, Verifying a static RAM design by **logic simulation**, Proceedings of the ... as well as how those limitations are overcome by state **transition simulation**. ...
portal.acm.org/ citation.cfm?id=103519&jmp=citings&dl=GUIDE&dl=ACM&CFID=11111111&CFTOK... - Similar pages

[PDF] Computing the Maximum Power Cycles of a Sequential Circuit
File Format: PDF/Adobe Acrobat
... easiest approach to power estimation is the one based on **logic simulation** It is ... unit delay The basic idea of the ADD-based symbolic **transition simulation** is as ...
dx.doi.org/10.1145/217474.217501 - Similar pages

[PS] A Methodology for Hardware Verification
File Format: Adobe PostScript - View as Text
... and mathematical background for describing system specifications, **logic simulation**, and the ... how these limitations are overcome by state **transition simulation**. ...
www.cs.cmu.edu/~bryant/pubdir/jacm91.ps - Similar pages

[PDF] Efficient and Fast Current Curve Estimation of CMOS Digital ...
File Format: PDF/Adobe Acrobat
... Input **transition Simulation** Engine Curent pulse calculator Cumulative graph generator ... signal quantisation is quite evident in **logic simulation** results, in part ...
www.springerlink.com/index/ETM69YGHJVA13XLH.pdf - Similar pages

In order to show you the most relevant results, we have omitted some entries very similar to the 4 already displayed.
If you like, you can repeat the search with the omitted results included.

"logic simulation" "transition simu" Search

Search within results | Language Tools | Search Tips | Dissatisfied? Help us improve

Google Home - Advertising Solutions - Business Solutions - About Google

Web

Tip: Try removing quotes from your search to get more results.

Your search - **"logic state simulation" "transition simulation"** - did not match any documents.

Suggestions:

- Make sure all words are spelled correctly.
- Try different keywords.
- Try more general keywords.
- Try fewer keywords.

Also, you can try Google Answers for expert help with your search.

Web

Tip: Try removing quotes from your search to get more results.

Your search - **"logic state simulation" "transient simulation"** - did not match any documents.

Suggestions:

- Make sure all words are spelled correctly.
- Try different keywords.
- Try more general keywords.
- Try fewer keywords.

Also, you can try Google Answers for expert help with your search.

Your search matched **0** of **1024576** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enteri new one in the text box.

(event <near/1> driven) <paragraph> (state <near/1 Search

Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

Results:
No documents matched your query.

Your search matched **0** of **1024576** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enter new one in the text box.

(event <near/1> driven) <and> (state <near/1> drive Search

Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

---

Results:
No documents matched your query.

Your search matched **3** of **1024576** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enteri new one in the text box.

(event <near/5> driven) <and> (state <near/5> drive Search

Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

---

1 **Logic simulation using networks of state machines**
Maurer, P.M.;
Design, Automation and Test in Europe Conference and Exhibition 2000. Proceedings , 27-30 March 2000
Pages:674 - 678

[Abstract] [PDF Full-Text (92 KB)] **IEEE CNF**

---

2 **State transformation in event driven explicit simulation**
Nguyen, T.V.; Devgan, A.;
Computer-Aided Design, 1997. Digest of Technical Papers., 1997 IEEE/ACM International Conference on , 9-13 Nov. 1997
Pages:289 - 294

[Abstract] [PDF Full-Text (488 KB)] **IEEE CNF**

---

3 **An innovative procedure for reliability assessment of power electr equipped systems: a real case study**
Fazio, V.; Savio, S.; Firpo, P.;
Industrial Electronics, 2000. ISIE 2000. Proceedings of the 2000 IEEE Interna Symposium on , Volume: 2 , 4-8 Dec. 2000
Pages:511 - 516 vol.2

[Abstract] [PDF Full-Text (656 KB)] **IEEE CNF**
